OIPE SEP 08 2006 PATENT & TRADEMARK OFFICE

FEE TRANSMITTAL for FY 2006

Patent fees are subject to annual revision.
Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12. See 37 C.F.R. §§ 1.27 AND 1.28

TOTAL AMOUNT OF PAYMENT ($) 620.00

Complete if Known	
Application Number	10/040,050
Filing Date	October 25, 2001
First Named Inventor	Mahesh S. Maddury
Examiner Name	William S. Powers
Group/Art Unit	2134
Attorney Docket No.	50325-0598 (Seq. No. 4395)

METHOD OF PAYMENT (check one)

1. ☒ Throughout the pendency of this application, please charge any additional fees, including any required extension of time fees, and credit all overpayments to deposit account 50-1302. A duplicate of this sheet is enclosed.

Deposit Account Number: 50-1302

Deposit Account Name: Hickman Palermo Truong & Becker, LLP

2. ☒ Payment Enclosed: ☒ Check ☐ Money Order ☐ Other

3. ☐ Applicant(s) is entitled to small entity status. See 37 CFR 1.27.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1011	300	2011	150	Utility filing fee	
1111	500	2111	250	Utility Search fee	
1311	200	2311	100	Utility Examination fee	
1081	250	2081	125	Utility Application Size Fee	
1005	200	2005	100	Provisional Application Fee	
1085	250	20835	125	Provisional Application Size Fee	

SUBTOTAL (1) ($) 0.00

2. EXTRA CLAIM FEES

	Highest Paid Claims		Extra Claims		Fee from Below		Fee Paid
Total Claims		-20**=	0	x	50.00	=	0.00
Independent Claims		- 3**=	0	x	200.00	=	0.00
Multiple Dependent						=	

**or number previously paid, if greater; For Reissues, see below

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	**Reissue independent claims over original patent
1205	50	2205	25	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge – late provisional filing fee or cover sheet.	
1251	120	2251	60	Extension for reply within first month	120.00
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	500.00
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1504	300	2504	300	Publication Fee	
1462	400	1462	400	Petitions Director not specifically provided for Group I	
1463	200	1463	200	Petitions Director not specifically provided for Group II	
1464	130	1464	130	Petitions Director not specifically provided for Group III	
1806	180	1806	180	Submission of information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
				Other fee (specify)	
				Other fee (specify)	

*Reduced by Basic Filing Fee Paid SUBTOTAL (3) ($) 620.00

SUBMITTED BY

Name (Print/Type)	Craig G. Holmes	Registration No. (Attorney/Agent)	44,770	Telephone	(408) 414-1080
Signature	[signature]			Date	9/1/06

WARNING: **Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

AF 2134

Docket No. 50325-0598 (Seq. No. 4395) PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Confirmation No.: 1826
Mahesh S. MADDURY, et al.	Group Art Unit: 2134
Serial No.: 10/040,050	Examiner: William S. Powers
Filed: October 25, 2001	
For: METHOD AND APPARATUS FOR CALCULATING A MULTIPLICATIVE INVERSE OF AN ELEMENT OF A PRIME FIELD	

APPEAL BRIEF

Hon. Commissioner for Patents
Mail Stop APPEAL BRIEF
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal that was filed on June 1, 2006, and based on which a one-month extension of time for filing of this Appeal Brief is hereby requested. The extension of time fee is enclosed.

I. REAL PARTY IN INTEREST

Cisco Systems, Inc., which owns the assignee Cisco Technology, Inc., both of San Jose, California, are the real parties in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1-16 are pending in the application and were finally rejected in the Final Office Action mailed on March 1, 2006. Specifically, Claims 5, 6, 7, 10, and 11 have been rejected

under 35 U.S.C. § 102(b) as allegedly unpatentable over U.S. Patent Number 5,414,772 issued to Naccache et al. ("*Naccache*"). Claims 1-4 and 12-16 have been rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over Naccache in view of the alleged "Applicant admitted prior art." It is from this final rejection of Claims 1-16 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

The claims have not been amended after the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claims 1, 2, 3, 4, 5, 12, 14, 15, and 16. The application and claims generally address problems with calculating a multiplicative inverse of a prime field for cryptography and digital signatures for secure computer networking. To summarize the following discussion, the approaches of the claims of the present application involve the use of a modulo exponentiation block to determine a multiplicative inverse based on using a prime modulus. The approaches of the claims can be implemented without the conventional extended Euclidean algorithm (EEA) of the prior art, which is an iterative algorithm for computing a multiplicative inverse, and which involves a modular inverter circuit for determining the multiplicative inverse.

A. Claims 1, 14, and 15

Independent Claims 1, 14, and 15 recite similar features, except in the contexts of a method, a computer-readable medium, and an apparatus with means plus function limitations, respectively. Claims 1, 14, and 15 are directed to generating a multiplicative inverse for use in determining a digital signature based upon using a modulo exponentiation block with a prime modulus in which the exponent is two less than the prime modulus. (Application, paragraph [0014], page 6, lines 4-7.)

Regarding **Claim 1**, the features therein correspond to an implementation of expression (23) as derived in the application, namely $a^{p-2} = a^{-1} \bmod p$, in which a modulo multiplicative inverse, $a^{-1} \bmod p$, is determined based on modulo exponentiation, $a^{p-2} \bmod p$, for p being a prime modulus. (Application, paragraphs [0041-0044], page 13, lines 2-19). The expression (23) is derived starting from Euler's Theorem, as shown in expression (21) of the Application, based on rewriting Euler's Theorem as in expression (22) and then restricting "b"

to be a positive prime number, p, thereby resulting in expression (23). In examining expression (23), it can be observed that the multiplicative inverse can be found in terms of modulo exponentiation based on a power of "p-2," provided that "p" is a prime number.

The approach of Claim 1 is implemented using "a modulo exponentiation block," such as modulo exponentiation (ME) block 120 of FIG. 1B, which avoids the problems of using the extended Euclidean algorithm (EEA) of the prior art. Note that the improvement of the approach of Claim 1 is at the "expense" of limiting/restricting "p" to being a prime modulus, not merely that "p" is relatively prime with respect to "a" as required by Euler's Theorem and as is implemented in the EEA. Also note that the EEA is an iterative approach and therefore is slow for large numbers, which is today common with key sizes of 1024, 2048, or even more bits. The prior art approaches for the EEA are implemented as a multiplicative inverse (MI) block through an application specific integrated circuits (ASICs), such as in hypothetical modulo multiplicative inverse (MI) block 190 of FIG. 1B, that occupy a large area of chip "real estate." (Application, paragraphs [0008 – 0011], page 3, line 19 – page 4, line 24.)

However, in the approach of Claim 1, existing blocks, such as a modulo exponentiation (ME) block, can be used that have smaller area requirements when implemented on a chip. (Application, paragraph [0009], page 4, lines 5-8.) While this improvement is at the "expense" of requiring that the modulus be a prime modulus, which is required in order to derive expression (23), such an expense is generally outweighed by the result of being able to calculate a multiplicative inverse using modular exponentiation in lieu of an ASIC that implements the larger and more time consuming MI circuitry. (*See* Application, paragraphs [0010-0011], page 4, lines 9-19.) In addition, by using an ME block, the developmental effort required to design, fabricate, testing, revise, and certify a new block, such as an MI block that is used to implement the EEA of the prior art, is avoided along with the associated significant increase in the time-to-market. (Application, paragraph [0011], page 4, lines 20-24.)

Specifically in Claim 1, the modulo exponentiator block is used in "determining a multiplicative inverse of the first integer data value modulo a prime modulus by computing a first quantity modulo the prime modulus data value." For example, the first integer data value is the value for which the multiplicative inverse is desired, such as "a" in expression (23) of the application. The prime modulus is "p" in expression (23).

Next in Claim 1, the "first quantity equals, modulo the prime modulus data value, the first integer data value raised to a power of a second quantity." For example, the first quantity is "a" in expression (23) modulo the prime modulus "p" raised to the power of the second quantity. Then in Claim 1, the "second quantity is two less than the prime modulus data value." For example, the second quantity is the exponent of expression (23), namely "p-2" or two less than the prime modulus.

The approach of Claim 1 can be used in a network, such as that illustrated in FIG. 1A, that includes digital-signature application specific integrated circuits (ASICs) 131 and 133 as part of gateway devices 130 and 132, respectively, that are utilized for secure communications between clients 110 and 112 that connect to gateways 130 and 132 via local networks 152 and 152, respectively. (Application, FIG. 1A, paragraphs [0052-0054], page 15, line 13 – page 16, line 16.) In particular, FIG. 2 is a block diagram of a portion 200 of an integrated circuit for generating a DSA digital signature that incorporates the use of two modulo exponentiation circuits, 220a and 220b, in addition to other circuits shown therein such as modulo multiplication circuits 240a and 240b, while FIG. 3 is a block diagram of a portion 300 of an integrated circuit for generating a digital signature that incorporates a single ME block 120 in lieu of the two ME blocks and two modular multiplication blocks of FIG. 2.

The approach of **Claim 14** is supported by the same portions of the application as described above with respect to Claim 1. Specifically, Claim 14 is a computer-readable medium claim that features the same steps as in Claim 1. In addition, Claim 14 being directed to a computer-readable medium that carries sequences of instructions that cause on one or more processors to carry out the recited steps is supported by the "HARDWARE OVERVIEW" section of the application.

In particular, main memory 506 of FIG. 5 stores instructions for execution by processor 504. (Application, paragraph [0101], page 26, lines 12-14; paragraph [0104], page 27, lines 15-17.) Such instructions can be read into main memory 506 from a computer-readable medium, such as storage device 510, in which execution of the sequences of instructions contained in main memory 506 causes the processor 504 to perform the process steps described in the application. (Application, paragraph [0101], page 26, lines 14-16; paragraph [0104], page 27, lines 18-20.) The term "computer-readable medium" refers to any medium that participates in providing instructions to processor 805 for execution, including

but not limited to, non-volatile media, volatile media, and transmission media. (Application, paragraph [0105]. page 28, lines 1-8.)

Regarding **Claim 15**, which is in the context of an apparatus, the above discussion with respect to Claim 1 applies as Claim 15 includes means plus function elements (identified by the elements that begin "means for...") for performing the functions of the steps recited in Claim 1. In particular, the approach of Claim 15 can be implemented on computer or a network device such as a router that includes a digital signature ASIC, such as digital signature ASIC 130 and 131 of FIG. 1A. (Application, paragraph [0052], page 15, lines 19-23.) The digital signature ASICs can include an ME block 120, as illustrated in FIG. 1B, one or more of which can be implemented in a portion of an integrated circuit, such as those illustrated in FIG. 2 and FIG. 3.

Specifically, the "means for receiving and storing a first integer data value" of Claim 15 includes digital signature ASICs 130 and 131 as implemented in FIGs. 2 and 3, such as through K input register 202 of FIG. 2 that feeds the base input 121 of ME block 220b, the input queue 312 of FIG. 3 that feeds the base input 121 of ME block 120, along with more general hardware and software implementations based on a computer system 500, such as that illustrated in FIG. 5. The "means for determining a multiplicative inverse of the first integer data value modulo a prime modulus data value by computing a first quantity modulo the prime modulus data value, wherein said computing includes using a modulo exponentiation block" of Claim 15 includes digital signature ASICs 130 and 131 as implemented in ME block 220b of FIG. 2, ME block 120 of FIG. 3, along with more general hardware and software implementations based on a computer system 500, such as that illustrated in FIG. 5. Finally, the "means for storing the multiplicative inverse in a computer hardware storage element for use in determining the digital signature of the electronic message" of Claim 15 includes digital signature ASICs 130 and 131 as implemented in FIGs. 2 and 3, such as output channel of ME blocks 220b and 120, along with more general hardware and software implementations based on a computer system 500, such as that illustrated in FIG. 5.

B. Claim 5

Independent Claim 5 is an apparatus claim from which Claims 6-11 depend. Claim 5 is directed to an apparatus for performing a particular operation for using digital signatures on a network. The discussion above with respect to Claims 1, 14, and 15 applies to Claim 5 as

well, although Claim 5 does not include all the features of Claims 1, 14, and 15. Specifically, Claim 5 is not limited to determining a digital signature but rather for use with digital signatures, such as RSA encrypting and decrypting operations, although such features are included in dependent Claims 8 and 9 that depend from Claim 5, along with digital signature algorithm signing and verifying operations, as in dependent Claims 10 and 11, respectively, that also depend from Claim 5.

The features of Claim 5 are based upon implementing expression (23) in which a modulo multiplicative inverse, a^{-1} mod p, is determined based on modulo exponentiation for p being a prime modulus. (Application, paragraphs [0041-0044], page 13, lines 2-19), although unlike Claims 1, 14, and 15, Claim 5 is not limited to the use of an exponent (e.g., "second quantity") that is equal to two less than the prime modulus. However, Claim 5 is similar to Claims 1, 14, and 15 in featuring the use of a modulo exponentiation block with a prime modulus to produce a multiplicative inverse of an integer, and thus those portions of the discussion of Claims 1, 14, and 15 apply equally to Claim 5.

C. Claims 2, 12, and 16

Independent Claims 2, 12, and 16 recite similar features, except in the contexts of a method, an apparatus, a computer-readable medium, and an apparatus with means plus function limitations, respectively. Claims 2, 12, and 16 are directed to generating an output signal indicating a multiplicative inverse of an integer data value modulo a prime modulus for use in performing a particular operation, in which a modulo exponentiation block is used along with an exponent that is two less than the prime modulus. (Application, paragraph [0014], page 6, lines 4-7.)

Regarding **Claim 2**, the discussion above with respect to Claims 1, 5, 14, and 15 apply equally to Claim 2, although the features of Claim 2 are in the form of sending three signals to the three inputs of the modulo exponentiation block. Specifically, the first signal indicates the value of the integer data value and is sent the base input of the modulo exponentiation block, such as the input K from K input register 202 of FIG. 2 that is sent to base input 121 of modulo exponentiation block 220b or the input K from input queue 311 that is sent to base input 121 of modulo exponentiation block 120 of FIG. 3.

The second signal indicates the value of the prime modulus and is sent to the modulus input of the modulo exponentiation block, such as the Q input register 204 of FIG. 2 that

sends Q to the modulus input 123 of modulo exponentiation block 220b or the input Q from input queue 313 that is sent to modulus input 123 of modulo exponentiation block 120 of FIG. 3.

The third signal indicates the value of the prime modulus less two and is sent to the modulus input of the modulo exponentiation block, such as exponent input 122 of modulo exponentiation block 220b of FIG. 2 or the value Q-2 from input queue 312 that is sent to exponent input 122 of modulo exponentiation block 120 of FIG. 3.

Then in Claim 2, the modulo exponentiation block generates the output based on the first quantity modulo a value at the modulus input in which the first quantity equals, modulo the value at the modulus input, a value at the base input raised to a power of a value at the exponent input, such as modulo exponentiation block 220b of FIG. 2 and modulo exponentiation block 120 of FIG. 3 that evaluate the multiplicative inverse based on the inputs, according to expression (23) of the Application.

Finally in Claim 2, the output of modulo exponentiation block is stored in a computer hardware storage element for using in performing a particular operation that is selected from the group consisting of a DSA signing operation, a DSA verifying operation, an encryption operation, and a decryption operation, such as the output of modulo exponentiation block 220b of FIG. 2 that is used as the second multiplier input 242 to modulo multiplication block 240b of FIG. 2 for evaluating expression (16) (see Application, paragraph [0074], page 21, lines 1-3) or modulo exponentiation block 120 of FIG. 3 at time t5 that is directed by switch 360 to the input queue 312, following which the fifth computation cycle evaluates expression (16) in modulo addition block 250.

The approach of **Claim 12** is supported by the same portions of the application as described above with respect to Claim 1. Specifically, Claim 12 is a computer-readable medium claim that features the same steps as in Claim 1. In addition, Claim 12 being directed to a computer-readable medium that carries sequences of instructions that cause on one or more processors to carry out the recited steps is supported by the "HARDWARE OVERVIEW" section of the application.

In particular, main memory 506 of FIG. 5 stores instructions for execution by processor 504. (Application, paragraph [0101], page 26, lines 12-14; paragraph [0104], page 27, lines 15-17.) Such instructions can be read into main memory 506 from a

computer-readable medium, such as storage device 510, in which execution of the sequences of instructions contained in main memory 506 causes the processor 504 to perform the process steps described in the application. (Application, paragraph [0101], page 26, lines 14-16; paragraph [0104], page 27, lines 18-20.) The term "computer-readable medium" refers to any medium that participates in providing instructions to processor 805 for execution, including but not limited to, non-volatile media, volatile media, and transmission media. (Application, paragraph [0105]. page 28, lines 1-8.)

Regarding Claim 16, which is in the context of an apparatus, the above discussion with respect to Claim 2 applies as Claim 16 includes means plus function elements (identified by the elements that begin "means for...") for performing the functions of the steps recited in Claim 2. In particular, as noted above, the means for sending the three signals of Claim 16 include the portions of the integrated circuits illustrated in FIGs. 2 and 3, as described above. Specifically, in FIG. 2, the K input register 202 sends the first signal to the base input 121 of modulo exponentiation block 220b, the Q input register 204 sends the second signal to the modulus input 123, the Q-2 register 209 sends the third signal to the exponent input 122, and the output channel of modulo exponentiation block 220b is uses as the second multiplier input 242 of modulo multiplication block 240b. Also, in FIG. 3, the first signal for the input K is sent by input queue 311 to base input 121 of modulo exponentiation block 120 at input time 4 374 as determined by switch 360, the second signal for the modulus Q is sent by input queue 313 to modulus input 123, the third signal for Q-2 is sent by input queue 312 to exponent input 122, and the output channel of modulo exponentiation block 120 is send to switch 360.

D. Claim 4

Independent Claim 4 is an apparatus claim that is directed to an apparatus for generating an output signal indicating a multiplicative inverse of an integer modulo a prime modulus. The discussion above with respect to Claims 2, 12, and 16 applies to Claim 4 as well.

The features of Claim 4 are based upon implemented expression (23) in a modulo exponentiation block, such as modulo exponentiation block 120 of FIG. 1B, modulo exponentiation block 220B of FIG. 2, and modulo exponentiation block 120 of FIG. 3. As noted on all three of those figures, each of the modulo exponentiation blocks has a base input, an exponent input, and a modulus input, and thus each of the modulo exponentiation blocks

implements expression (23) as described above with respect to Claims 2, 12, and 16, and thus the discussion of Claims 2, 12, and 16 apply equally to Claim 4.

E. Claim 3

Independent Claim 3 is a method claim for fabricating an electronic circuit that generates an output signal indicating a multiplicative inverse of an integer data value modulo a prime modulus. The discussion above with respect to Claims 2, 4, 12, and 16 applies to Claim 3, although Claim 3 includes steps of connecting three registers, which correspond to the three signals of Claims 2, 4, 12, and 16, to the three different inputs of a modulo exponentiation block, such as modulo exponentiation block 120 of FIG. 1B, modulo exponentiation block 220B of FIG. 2, and modulo exponentiation block 120 of FIG. 3.

The features of Claim 3 are based upon implementing expression (23) in which a modulo multiplicative inverse, a^{-1} mod p, is determined based on modulo exponentiation for p being a prime modulus. (Application, paragraphs [0041-0044], page 13, lines 2-19), by connecting the three registers to the three inputs of the modulo exponentiation block. Thus, the discussion of Claims 2, 4, 12, and 16 apply equally to Claim 5.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Rejections of Claims 5, 6, 7, 10, and 11 under 35 U.S.C. § 102(B)

Claims 5, 6, 7, 10, and 11 have been rejected under 35 U.S.C. § 102(b) as allegedly unpatentable over U.S. Patent Number 5,414,772 issued to Naccache et al. ("*Naccache*").

B. Rejections of Claims 1-4 and 12-16 under 35 U.S.C. § 103(C)

Claims 1-4 and 12-16 have been rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over *Naccache* in view of the alleged "Applicant admitted prior art."

VII. ARGUMENT

A. Introduction

(1) CLAIMS 5, 6, 7, 10, AND 11 ARE PATENTABLE OVER *NACCACHE*

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." MPEP §2131, citing *Verdegaal Bros., Inc. v. Union Oil Co.*, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The Appellants respectfully submit that *Naccache* fails to expressly or inherently disclose all of the limitations of Claims 5, 6, 7, 10, and 11. In particular, *Naccache* fails to disclose (1) "a **modulo exponentiation block**"; (2) that "a modulo exponentiation block is **configured for producing a multiplicative inverse** of an integer;" and (3) the "multiplicative inverse of an integer" produced by the modulo exponentiation block is "**modulo a prime modulus**." Therefore, the Appellants respectfully submit that the Examiner has failed to establish that Claims 5, 6, 7, 10, and 11 are anticipated because *Naccache* fails to expressly or inherently disclose all of the limitations of Claims 5, 6, 7, 10, and 11.

(2) CLAIMS 1-4 AND 12-16 ARE PATENTABLE OVER *NACCACHE* IN VIEW OF THE ALLEGED "APPLICANT ADMITTED PRIOR ART"

"To establish a prima facie case of obviousness [under 35 U.S.C. §103(a)] of a claimed invention, all the claim limitations must be taught or suggested by the prior art." MPEP §2143.03. In addition, a sufficient factual basis to support the obviousness rejection must be proffered. *In re Freed*, 165 USPQ 570 (CCPA 1970); *In re Warner*, 154 USPQ 173 (CCPA 1967); *In re Lunsford*, 148 USPQ 721 (CCPA 1966). In particular, neither *Naccache* nor the alleged "Applicant admitted prior art," either alone or in combination, disclose (1) "a **modulo exponentiation block**;" (2) that "a modulo exponentiation block" is used in "**determining a multiplicative inverse** of the first integer data value;" (3) the "multiplicative inverse of the first integer data value" is determined "by computing a first quantity **modulo the prime modulus** data value;" and (4) "wherein the first quantity equals, modulo the prime modulus data value, the first integer data value *raised to a power of a second quantity*" "wherein the *second quantity* is **two less than the prime modulus** data value." The Appellants respectfully submit that neither *Naccache* nor the alleged "Applicant admitted

prior art," either alone or in combination, disclose, teach, suggest, or render obvious all the limitations of Claims 1-4 and 12-16. Therefore, the Appellants respectfully submit that the Examiner has failed to establish a prima facie case of obviousness because many of the claim limitations are neither taught nor suggested by the prior art.

B. Claim 5 is Patentable over *Naccache*

(1) INTRODUCTION TO CLAIM 5

Claim 5 features:

> "An apparatus for performing a particular operation for using digital signatures on a network, the apparatus comprising **a modulo exponentiation block** *configured for* **producing a multiplicative inverse** of an integer modulo a **prime modulus,** wherein said multiplicative inverse is used in performing the particular operation." (Emphasis added.)

Thus, Claim 5 features (1) "a modulo exponentiation block," (2) that the modulo exponentiation block is "configured for producing a multiplicative inverse of an integer," and (3) that the multiplicative inverse is "modulo a prime modulus."

(2) INTRODUCTORY DISCUSSION OF *NACCACHE*

In contrast to the approach of Claim 5, *Naccache* discloses a system for improving the digital signature algorithm (DSA), and in particular, a system to compute "the inverse of a first number x modulo a second number n and use the resulting modular inverse" in a cryptographic protocol. (Title, Abstract.) *Naccache's* invention is directed to improving such protocols when "modular inverses are to be computed by a slow apparatus connected to a communication link to a faster but potentially hostile device." (Col. 1, lines 6-11.) *Naccache* explains that "the calculation of a modular inverse (1/k mod q), can be done either with a lengthy series of modular multiplications or with Euclidean inversion involving long integer division. (Col. 2, lines 34-50.) The latter is a reference to the EEA described previously herein.

Naccache then describes the approach described therein as being directed to allowing an apparatus A, such as a smart card, badge, or electronic key, to use the computational power

of a potentially hostile device B, such as a PC, vending machine, or access-control gate to compute modular inverses. (Col. 2, lines 56-62.) *Naccache* explains that this is accomplished by exchanging data between apparatuses A and B, as illustrated in Figure 4, which involves the use of a random number generator by apparatus A. (Col. 4, lines 35-39.) As a result, *Naccache* states that "the computation of the modular inverse is traded-off against one additional random or pseudo random number generation, two modular multiplications and the transmission of one number," and therefore that because "the communication or data takes negligible time when compared to the effort requested for computing modular inverses," *Naccache's* approach "offers advantageous trade-off possibilities versus prior state [of the] art." (Col. 6, lines 12-24.)

In essence, *Naccache* is implementing extra overhead via the random number generation and data communication between apparatuses A and B to allow the faster processing apparatus B to determine a multiplicative inverse needed by apparatus A. However, the computation of the multiplicative inverse by apparatus B is nothing more than either the use of many modular multiplications or the EEA, since *Naccache* merely notes that apparatus B computes the multiplicative inverse z that is the inverse of y modulo n. (*See* Col. 4, lines 45-49.) Thus, other than which apparatus determines the multiplicative inverse, there is nothing new or different in *Naccache's* approach for the calculation of the multiplicative inverse itself as compared to the prior art approach of the EEA.

(3) *NACCACHE* DOES NOT DISCLOSE "A MODULO EXPONENTIATION (ME) BLOCK"

The Final Office Action states that "Naccache teaches determining a multiplicative inverse of a first integer value, y, through the processing means (column 4, lines 45-48) that include an exponentiation circuit, modular inversion circuit as well as other circuits (figure 2)."

Naccache states that Figures 1 and 2 are block diagrams of apparatus A and apparatus B, respectively. (Col. 3, lines 27-30; noting that the description of Figure 2 includes a typographical error instead of "B".) A review of Figure 2 shows that apparatus B includes a "Powerful C.P.U." 30 that includes means for performing the following operations: "multiplication;" "modular inversion;" "exponentiation;" "modular reduction;" and

"comparing." As described in *Naccache's* specification, Figure 2 is illustrating that the "CPU and/or ROM of apparatus B have stored therein programs or computational resources corresponding to or implementing...multiplication, modular inversion, comparing, exponentiation and modular reduction. Some of these operations can be merged together (for instance, the modular reduction can be integrated into the multiplication)."

Note that in illustrating and describing Figure 2, *Naccache* is careful to note which operations are "modular" (e.g., modular inversion and modular reduction) and which are not modular (e.g., multiplication, exponentiation, and comparing). This means that the exponentiation illustrated and described in Figure 2 of *Naccache* is ***<u>not modular</u>***. The same distinction between modular and non-modular operations can be seen in Figure 1, which also illustrates normal exponentiation that is ***<u>not modular</u>***. However, Claim 5 features "**a modular exponentiation block**," which by the name itself **means that the exponentiation is *<u>modular</u>*** in contrast to an exponentiation that is not modular, such as is depicted in Figures 1 and 2 of *Naccache*.

During the Interview conducted with the Examiner on May 2, 2006, the Examiner explained the reliance on *Naccache* and Figure 2 as follows: because *Naccache* says that "Some of these operations can be merged together (for instance, the modular reduction can be integrated into the multiplication)" (*see* Col. 4, lines 31-31) and that Figure 2 discloses a program for "multiplication" and a program for "modular reduction," and then that means that *Naccache* has disclosed a program for performing "modular multiplication." The Examiner then stated that the suggestion to combine elements of Figure 2 means that one can combine the program for "exponentiation" with the program for "modular reduction" as illustrated in Figure 2 to arrive at a program for "modular exponentiation," which is then the equivalent to a "modulo exponentiation block" as featured in Claim 5. However, the Examiner's reasoning is incorrect, for several reasons.

First, nowhere in *Naccache* is there any teaching or suggestion, either express or implied, that combining the program for "multiplication" and the program for "modular reduction" of Figure 2 results in program for "modular multiplication." *Naccache's* suggestion that the operations of Figure 2 can be combined is most reasonably interpreted as a generalizing statement that the programs for performing the operations illustrated within

CPU 30, although illustrated as being separate, need not be separate programs or separate operations.

Second, even if the "multiplication" and "modular reduction" programs/operations were combined, that would be understood to mean that the multiplication operations would be performed, followed by a modular reduction of the result. While that result may be the same as performing modular multiplication, a program for doing modular multiplication would be performed using modular mathematics, not normal non-modular mathematics followed by a modular reduction of a non-modular result. As a specific example, multiplying 4 times 5 results in 20, and based on a modulus of 8, taking a modular reduction of 20 mod 8 results in 2. However, the modular multiplication of 4 times 5 directly provides the result 2.

Third, even if one were accept for the moment the Examiner's premise that a program for multiplication when combined with a program for modular reduction somehow results in a program capable of modular multiplication (and ignoring for the moment that this would mean the rejection of Claim 5 based on *Naccache* is no longer properly characterized as being under 102(b) and would instead be an obviousness rejection under 103(a)), there is no teaching, suggestion, or motivation to combine the "exponentiation" program with the "modular reduction" program of Figure 2 of *Naccache*, other than the hindsight of trying to arrive at the "modulo exponentiation block" of Claim 5, which the Appellants respectfully submit is impermissible hindsight based solely on the Appellant's own claims.

Therefore, while *Naccache* does disclose programs for a CPU for performing exponentiation and modular reduction, the Appellants respectfully submit that *Naccache* fails to disclose "a modulo exponentiation block" as featured in Claim 5.

(4) *NACCACHE* DOES NOT DISCLOSE AN ME BLOCK CONFIGURED TO PRODUCE A MULTIPLICATIVE INVERSE

Next, assuming for this argument only that *Naccache's* disclosure of programs for a CPU for performing exponentiation and modular reduction is interpreted to be a disclosure of "a modulo exponentiation block," *Naccache* still fails to disclose "a modulo exponentiation block **configured for producing a multiplicative inverse**," as featured in Claim 5.

In particular, Figure 2 of *Naccache* illustrates "modular inversion" as being separate from the other programs of powerful CPU 30. Recall that the basic idea of *Naccache's*

approach is to have the powerful CPU of apparatus B , as illustrated in Figure 2, determining a multiplicative inverse instead of the weak CPU of apparatus A, as illustrated in Figure 1, determining the multiplicative inverse (hence why Figure 1 does not illustrate a program for "modular inversion" for CPU 11). But if one accepts the proposition of the Examiner that combining the "exponentiation" program and the "modular reduction" program of CPU 30 results in "modular exponentiation" program, that resulting "modular exponentiation" program would still not be "configured for producing a multiplicative inverse of an integer." Rather, that resulting "modular exponentiation" program would just produce a modular exponentiation result.

Furthermore, even if one were to accept that such a "modular exponentiation" program formed by combining the "exponentiation" program and "modular reduction program of Figure 2 is a proper interpretation of *Naccache*, there would still be no motivation to use such a "modular exponentiation" program to produce a multiplicative inverse ***because Figure 2 already includes a program for*** "**modular inversion**." To accept the Examiner's argument would mean fundamentally altering the operation of powerful CPU 30 by having the multiplicative inverse produced by the hypothetical "modular exponentiation" program instead of by the *expressly illustrated and described* "modular inversion" program of Figure 2. This would also mean that the "modular inversion" program of Figure 2 would be irrelevant since there would be no need to use that program if one were to use the Examiner's hypothetical "modular exponentiation" program.

Therefore, even if *Naccache* is interpreted as suggesting a "modular exponentiation" program, the Appellants respectfully submit that *Naccache* fails to disclose "a modulo exponentiation block configured for producing a multiplicative inverse," as featured in Claim 5, because Naccache already expressly discloses a program for "modular inversion."

(5) *NACCACHE* DOES NOT DISCLOSE A MULTIPLICATIVE INVERSE PRODUCED BY AN ME BLOCK THAT IS MODULO A PRIME MODULUS

Next, assuming for this argument only that *Naccache* does disclose "a modulo exponentiation block configured for producing a multiplicative inverse of an integer," *Naccache* still fails to disclose "a modulo exponentiation block configured for producing a multiplicative inverse of an integer **modulo a prime modulus**." Nowhere in *Naccache* is

there a suggestion, teaching, or disclosure of the use of a prime modulus to determine a multiplicative inverse using a modulo exponentiation block.

During the Interview conducted on May 2, 2006, the Appellants pointed out to the Examiner that the portions of *Naccache* cited in the Final Office Action disclose nothing about a prime modulus, although the Appellants did note during the Interview that other portions of *Naccache* did refer to a prime modulus, such as Col. 1, lines 55-56 and Col. 5, lines 41-42 that described the DSA algorithm. The Examiner confirmed that the Final Office Action was based on those other portions of *Naccache* that disclosed a prime modular in rejecting Claim 5, although those portions of Naccache were not cited in the Final Office Action.

However, even given the disclosure of a prime modulus in the description of the DSA algorithm within *Naccache*, the discussion of the calculation of the multiplicative inverse by *Naccache* says nothing about the use of a prime modulus. Specifically, *Naccache* explains that "Apparatus B computes, by its processing means 30, a number z such that $z\,y = 1 \bmod n$ (that is, z is the inverse of y modulo n) and returns z to the apparatus A via the communication interface 31." (Col. 4, lines 40-44.) However, there is nothing in this portion of *Naccache* that n is a prime modulus. Even in the context of DSA, which *Naccache* explains involves the prime modulus p, the multiplicative inverse is not based on modulus p but rather q. (*See* Col. 2, lines 19-20.) Thus, the only motivation for generating a multiplicative inverse with a modulo exponentiation block using a prime modulus is the hindsight observation based on the Appellants' own claims, which the Appellants again respectfully submit is improper.

Therefore, even if *Naccache* is interpreted as suggesting a that *Naccache* does disclose "a modulo exponentiation program" that is configured for producing a multiplicative inverse of an integer, the Appellants respectfully submit that *Naccache* fails to disclose "a modulo exponentiation block configured for producing a multiplicative inverse of an integer modulo a prime modulus," as featured in Claim 5.

(6) CONCLUSION OF DISCUSSION OF CLAIM 5 AND *NACCACHE*

Because *Naccache* fails to disclose, either expressly or inherently, (1) "a **modulo exponentiation block**"; (2) that "a modulo exponentiation block is **configured for producing a multiplicative inverse** of an integer;" and (3) the "multiplicative inverse of an

integer" produced by the modulo exponentiation block is "**modulo a prime modulus**," the Appellants respectfully submit that, for at least the reasons stated above, Claim 5 is allowable over the art of record and is in condition for allowance.

C. Claim 6 is Patentable over *Naccache*

Claim 6 depends from Claim 5 and features that "the apparatus has no circuitry block configured to perform an extended Euclidean algorithm (EEA) and no general-purpose processor configured by instructions to perform the EEA." The Final Office Action states that Naccache does not teach the use of the Extended Euclidean Algorithm (EEA) or the configuration of circuits to perform EEA operations (column 1, line 1 – column 65, line 27)."

However, as noted above in the introductory discussion of *Naccache*, there is a reference to the EEA in *Naccache* when discussion modular inverses, specifically that "Euclidian inversion, involving long integer division is generally used." (Col. 2, lines 44-50.) Therefore, the Appellants respectfully submit that Claim 6 is allowable over the art of record and is in condition for allowance.

D. Claim 7 is Patentable over *Naccache*

Claim 7 depends from Claim 5 and features that "the particular operation is performed in a series of sequential computations accomplished over a corresponding series of computation cycles" and "the apparatus further comprises **connections** configured to use the modulo exponentiation block during a plurality of computation cycles of the series of computation cycles." (Emphasis added.) For example, the modulo exponentiation block can be incorporated into circuit such as illustrated in FIG. 3 of the Application in which modulo exponentiation block 120 is connected to input queues 311, 312, 313, and 314 that are examples of "**connections** configured to use the modulo exponentiation block," such as determined by switch 360 at the input times 371, 372, 373, 374, and 375. As a specific example, at time t3 or input time 373, the value 0 residing within input queue 314 signifies that multiplication is to be performed instead of exponentiation, and similarly at time t5 or input time 375. Thus, in the example of FIG. 3 of the Application, the modular exponentiation block 120 is used for both modular multiplication, modular exponentiation, and modular exponentiation to determine a multiplicative inverse.

The Final Office Action states that "Naccache teaches that the multiplicative inverse is found through a series of steps (column 4, lines 35-61)." However, this citations from the Final Office Action shows that the Examiner has incorrectly characterized Claim 7 because there is nothing in Claim 7 about the multiplicative inverse being found through a series of steps. Rather, Claim 7 features that apparatus further comprises "**connections**" that are configured for a particular purpose, namely to "use the modulo exponentiation block during a plurality of computation cycles of the series of computation cycles." Yet the cited portion of *Naccache* says nothing about any "connections," little less connections configured as in Claim 7. Rather, the cited portion of *Naccache* merely outlines the approach for using apparatus B to determine a modular inverse required by apparatus A.

Therefore, the Appellants respectfully submit that Claim 7 is allowable over the art of record and is in condition for allowance.

E. Claim 1 is Patentable over *Naccache* in View of the Alleged "Applicant Admitted Prior Art"

(1) INTRODUCTION TO CLAIM 1

Claim 5 features:

"A data processing method for generating a multiplicative inverse for use in determining a digital signature, the method comprising the computer-implemented steps of:

receiving and storing a first integer data value relating to a digital signature of an electronic message;

determining a multiplicative inverse of the first integer data value **modulo a prime modulus** data value by computing a first quantity modulo the prime modulus data value, wherein said computing includes using a **modulo exponentiation block**;

wherein the first quantity equals, modulo the prime modulus data value, the first integer data value *raised to a power of a second quantity*;

wherein the *second quantity* is **two less than the prime modulus** data value; and

storing the multiplicative inverse in a computer hardware storage element for use in determining the digital signature of the electronic message." (Emphasis added.)

Thus, Claim 1 features (1) "a modulo exponentiation block," (2) that the modulo exponentiation block is used for "determining a multiplicative inverse," (3) that the multiplicative inverse is "modulo a prime modulus," and (4) the exponent used is "two less than the prime modulus."

(2) INCORPORATION OF ARGUMENTS FOR CLAIM 5 INTO ARGUMENTS FOR CLAIM 1

Because Claim 1 features (1) "a modulo exponentiation block," (2) that the modulo exponentiation block is used for "determining a multiplicative inverse," and (3) that the multiplicative inverse is "modulo a prime modulus," which are either the same as or very similar to those of Claim 1, the arguments presented above with respect to Claim 5 apply equally to Claim 1. In addition, the Appellants provide the following additional arguments.

(3) DISCUSSION OF CLAIM 1 AND NACCACHE

The Final Office Action states that *Naccache* teaches "determining that x is the multiplicative inverse of the first integer value, y, through the processing means (column 4, lines 45-48) that include an exponentiation circuit, modular inversion circuit as well as other circuits (figure 2)." While the Final Office Action is correct in characterizing that the cited portions of *Naccache* disclose a modular inversion operation/program and an exponentiation operation/program (although the Appellants notes that these are not discloses as "circuits" as stated in the Final Office Action), Claim 1 features a modulo exponentiation block, which is not the same as the normal non-modular exponentiation block in the cited portions of *Naccache*, for the reasons given above.

Furthermore, the Final Office Action's description of the cited portions of *Naccache* is inconsistent, since the Final Office Action cites the "modular inversion" operation/program along with the "exponentiation" operation/program (that the Examiner has mistakenly interpreted as a "modulo exponentiation block," for the reasons given above). Because *Naccache* discloses a "modular inversion" program/operation, there is no disclosure, teaching, or suggestion of calculating a multiplicative inverse using anything other than that "modular inversion" program/operation. If one accepts that *Naccache* discloses a "modulo

exponentiation block" based on the non-modular "exponentiation" program/operation of Figure 2, then there is no need for the "modular inversion" program/operation. Since that would fundamentally alter the operation of apparatus B in *Naccache's* approach and render the "modular inversion" program/operation of Figure 2 irrelevant, it is clear that the Examiner's reliance on the non-modular "exponentiation" program/operation as disclosing the "modulo exponentiation block" of Claim 1 is incorrect.

(4) THE ALLEGED "APPLICANT ADMITTED PRIOR ART"

The Final Office Action identifies the alleged "Applicant admitted prior art" referred to in the Final Office Action as follows: "in an analogous art, Euler's Theorem teaches that the exponent is two less than the prime modulus in a multiplicative inverse calculation (paragraph 43 of Specification)." However, Euler's Theorem teaches nothing about the exponent being two less than the prime modulus in a multiplicative inverse calculation. The cited portion of the Applicant's specification is as follows:

"According to **Euler's theorem**, for two positive numbers a and b that are relatively prime,

$$a^{\phi(b)} = 1 \bmod b \tag{21}$$

This expression can be rewritten as

$$a^{\phi(b)-1} = a^{-1} \bmod b \tag{22}$$

For ***b equal to a positive prime number p***, this expression becomes

$$a^{p-2} = a^{-1} \bmod p \qquad (23)\text{"}$$

(Emphasis added.)

This portion of the Applicant's specification shows that ***the Examiner is relying <u>not</u> upon Euler's Theorem*** for the feature "the second quantity is two less than the prime modulus data value," but rather the Examiner is instead impermissibly relying upon Applicant's derived expression (23) that shows the exponent "**p-2.**" Clearly, expression (23) is not Euler's Theorem since expression (23) requires that b be a prime number p which is not a requirement of Euler's Theorem. Furthermore, because the Appellants have not stated that expression (23) is "prior art" and rather would characterize expression (23) as most certainly

not being prior art (see MPEP §2129), the Examiner has incorrectly characterized expression (23) as "Applicant's admitted prior art."

Note that Euler's Theorem states a relationship, illustrated by expression (21), for two particular positive numbers a and b that are "relatively prime." Also note that "relatively prime" means that the only common denominator, sometimes referred to as the "greatest common denominator" or "gcd," between the two numbers is 1. However, Euler's Theorem does not require either a or b to themselves be prime. For example, the numbers 4 and 9 are relatively prime because the greatest common denominator between 4 (with denominators of 1, 2, and 2) and 9 (with denominators 1, 3, and 3) is 1. Yet 4 is clearly not prime because the factors of 4 are 1, 2, and 2, with 2 being repeated, and 9 is also clearly not prime because the factors of 9 are 1, 3, and 3, with 3 being repeated. Thus, choosing a and b to be 4 and 9, respectively, satisfies the requirements of Euler's Theorem and therefore expression (21) holds for 4 and 9. However, expression (23) does not apply to a and b equal to 4 and 9, respectively, because b is not a prime number, as required by expression (23).

While the Appellants do not claim to have invented Euler's Theorem, or expression (21), the Appellants disagree with characterizing either expression (22) or expression (23) as "prior art." Expression (22), which is a manipulation of expression (21), is not Euler's Theorem because expression (22) relies upon modular exponentiation to determine a multiplicative inverse, which is unlike Euler's Theorem of expression (21) that does not include modular exponentiation to determine a multiplicative inverse.

Furthermore, in deriving expression (23), the Appellants have introduced a further restriction not present or required by Euler's Theorem, namely that b is equal to a positive ***prime*** number p. Recall that Euler's Theorem merely requires b to be positive and relatively prime with respect to each other, but neither is required to be prime. Yet in deriving expression (23), a new limitation is introduced, namely that b is prime. Thus, the example above for a and b equal to 4 and 9, respectively, no longer can be considered an example of expression (23) since b equal to 9 is not a prime number. However, if b were 11 instead of 9, then expression (23) is valid because 11 is prime, along with being positive and relatively prime with respect to a value of 4 for a.

Note that is only when b is prime as required by expression (23), and not just relatively prime with respect to a as required by Euler's Theorem of expression (21), that the totient

function $\phi(b)$ is equal to "p-1," and thus $\phi(b)-1$ becomes "p-2," as shown in expression (23). As a result of expression (23), it can be seen that the multiplicative inverse can be obtained through the use of modular exponentiation, which is not the case with Euler's Theorem of expression (21).

Fortunately, during the Interview with the Examiner on May 2, 2006, the Examiner did agree that expression (23) of the Appellants specification is not Euler's Theorem and not "Applicant admitted prior art." However, during the Interview, the Examiner provided a new rationale, namely that expression (23) is a "special case" of Euler's Theorem. The Examiner then explained that because when the modulus b is restricted to being a prime modulus, Euler's Theorem becomes the same as expression (23), expression (23) is taught by Euler's Theorem, including the exponent being two less than the prime modulus.

The Appellants respectfully submit that the Examiner is in error in considering expression (23) to be a special case of Euler's Theorem, and therefore that Euler's Theorem discloses an exponent equal to two less than the prime modulus. Essentially, the Examiner's position is then that expression (23) is "prior art" as a special case of Euler's Theorem, which is contrary to the Examiner's statement during the Interview that expression (23) is not prior art.

Furthermore, one can clearly see from expression (21) that there is no modular exponentiation, little less a power of "p-2," little less the use of a prime modulus. While Euler's Theorem certainly applies to either a or b being prime, Euler's Theorem does not require, teach, or suggest that either be prime; rather, Euler's Theorem only requires that a and b are relatively prime. This means that the Examiner's "special case" argument essentially becomes the following: given expression (23), one can see that Euler's Theorem covers that expression, provided that b is limited to being a prime modulus. Yet the only motivation to make such an observation is the Appellant's own claims, which the Appellant's respectfully submit is not proper since the Appellants have not admitted that either expression (22) or expression (23) are prior art. Thus, the Examiner's argument is basically that the Examiner can use the Applicant's own disclosure to reject features of those claims, which is improper.

In addition, the "special case" rationale provided by the Examiner would mean that any invention based on a derivation from a known prior art starting point would be merely a "special case" of that known prior art starting point. The only inventions that would therefore

be patentable would be those that start from a completely novel starting point, which the Appellants respectfully submit would render a number of issued patents invalid, as many inventions are improvements based upon previous inventions. In addition, an improvement upon a previous invention could always be characterized as a "special case" of the prior invention, thereby allowing the prior invention to be used as prior art against the improvement because the latter is merely a "special case" of the former, which is absurd.

Therefore, the Appellants respectfully submit that Claim 1 is allowable over the prior art because the prior art fails to disclose, teach, suggest, or render obvious that "the second quantity is two less than the prime modulus data value" because the only disclosure, teaching, or suggestion of such a feature is from expression (23), which is not "prior art" and the only admitted prior art is Euler's Theorem, which says nothing about a prime modulus, little less an exponent equal to two less than the prime modulus, as in Claim 1.

(5) CONCLUSION OF DISCUSSION OF CLAIM 1, NACCACHE, AND THE ALLEGED "APPLICANT ADMITTED PRIOR ART"

Because *Naccache* and Euler's Theorem, either along or in combination, fail to disclose (1) "a **modulo exponentiation block**;" (2) that "a modulo exponentiation block" is used in "**determining a multiplicative inverse** of the first integer data value;" (3) the "multiplicative inverse of the first integer data value" is determined "by computing a first quantity **modulo the prime modulus** data value;" and (4) "wherein the first quantity equals, modulo the prime modulus data value, the first integer data value *raised to a power of a second quantity*" "wherein the *second quantity* is **two less than the prime modulus** data value," the Appellants respectfully submit that, for at least the reasons stated above, Claim 1 is allowable over the art of record and is in condition for allowance.

F. CLAIMS 14 AND 15 ARE PATENTABLE OVER *NACCACHE* IN VIEW OF EULER'S THEOREM

Claims 14 and 15 contain features that are the same as those described above with respect to Claim 1, although in the context of a computer-readable medium and an apparatus, respectively. In particular, both Claims 14 and 15 feature (1) "a **modulo exponentiation block**;" (2) that "a modulo exponentiation block" is used in "**determining a multiplicative**

inverse of the first integer data value;" (3) the "multiplicative inverse of the first integer data value" is determined "by computing a first quantity **modulo the prime modulus** data value;" and (4) "wherein the first quantity equals, modulo the prime modulus data value, the first integer data value *raised to a power of a second quantity*" "wherein the *second quantity* is **two less than the prime modulus** data value," which are the same as in Claim 1. Therefore, based on at least the reasons stated above with respect to Claim 1, the Appellants respectfully submit that Claims 14 and 15 are allowable over the art of record and is in condition for allowance.

G. Claims 2, 4, 12, and 16 are Patentable over *Naccache* in View of Euler's Theorem

Claims 2, 4, 12, and 16 contain features that are the same as or similar as those described above with respect to Claim 1. In particular, each of Claims 2, 4, 12, and 16 feature (1) "a **modulo exponentiation block**;" (2) that "the modulo exponentiation block generates an output" "indicating a **multiplicative inverse** of an integer data value" (3) the multiplicative inverse determined "based on a first quantity modulo a value at the modulus input" in which the "modulus input" is "a value of the **prime modulus**;" and (4) "wherein the first quantity equals, modulo the value at the modulus input, a value at the base input raised to a power of a value at the exponent input" that is "a value of the **prime modulus less two**," similar to Claim 1. Therefore, based on at least the reasons stated above with respect to Claim 1, the Appellants respectfully submit that Claims 2, 4, 12, and 16 are allowable over the art of record and is in condition for allowance.

H. Conclusion and PRAYER for Relief

Based on the foregoing, the Appellants respectfully submits that both (a) the rejections of Claims 5, 6, 7, 10, and 11 under 35 U.S.C. § 102(b) as allegedly unpatentable over *Naccache* and (b) the rejections of Claims 1-4 and 12-16 under 35 U.S.C. § 103(a) over *Naccache* in view of the alleged "Applicant admitted prior art" lacks the requisite factual and legal bases. The Appellants respectfully submit that the imposed rejections under 35 U.S.C. § 102(b) over *Naccache* and under 35 U.S.C. § 103(a) over *Naccache* in view of the alleged "Applicant admitted prior art" are **not** viable and respectfully solicit the Honorable Board to **reverse** each of the imposed rejections under 35 U.S.C. § 102(b) and 35 U.S.C. § 103(a).

To the extent necessary to make this reply timely filed, the Applicant petitions for an extension of time under 37 C.F.R. § 1.136.

If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Craig G. Holmes
Reg. No. 44,770

Date: September 1, 2006

2055 Gateway Place, Suite 550
San Jose, CA 95110-1089
Telephone: (408) 414-1207
Facsimile: (408) 414-1076

Claims Appendix

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Hon. Commissioner for Patents, Mail Stop APPEAL BRIEF, P.O. Box 1450, Alexandria, VA 22313-1450.

on September 1, 2006 by Tracy Reynolds

VIII. CLAIMS APPENDIX

1. (Previously Presented) A data processing method for generating a multiplicative inverse for use in determining a digital signature, the method comprising the computer-implemented steps of:

   receiving and storing a first integer data value relating to a digital signature of an electronic message;

   determining a multiplicative inverse of the first integer data value modulo a prime modulus data value by computing a first quantity modulo the prime modulus data value, wherein said computing includes using a modulo exponentiation block;

   wherein the first quantity equals, modulo the prime modulus data value, the first integer data value raised to a power of a second quantity;

   wherein the second quantity is two less than the prime modulus data value; and

   storing the multiplicative inverse in a computer hardware storage element for use in determining the digital signature of the electronic message.

2. (Previously Presented) A method for generating an output signal indicating a multiplicative inverse of an integer data value modulo a prime modulus for use in performing a particular operation, the method comprising the steps of:

   sending a first signal, indicating a value of the integer data value, to a base input of a modulo exponentiation block of an electronic integrated circuit;

   sending a second signal, indicating a value of the prime modulus, to a modulus input of the modulo exponentiation block; and

   sending a third signal, indicating a value of the prime modulus less two, to an exponent input of the modulo exponentiation block;

   wherein the modulo exponentiation block generates an output based on a first quantity modulo a value at the modulus input;

   wherein the first quantity equals, modulo the value at the modulus input, a value at the base input raised to a power of a value at the exponent input; and

wherein the output generated by the modulo exponentiation block is stored in a computer hardware storage element for use in performing a particular operation that is selected from the group consisting of a digital signature algorithm signing operation, a digital signature algorithm verifying operation, an encryption operation for a first electronic message, and a decryption operation for a second electronic message.

3. (Previously Presented) A method for fabricating an electronic circuit that generates an output signal indicating a multiplicative inverse of an integer data value modulo a prime modulus, the method comprising the steps of:
connecting a first register holding signals indicating a value of the integer data value to a base input of a modulo exponentiation block;
connecting a second register holding signals indicating a value of the prime modulus, to a modulus input of the modulo exponentiation block;
connecting a third register holding signals indicating a value of the prime modulus less two, to an exponent input of the modulo exponentiation block;
wherein the modulo exponentiation block generates an output based on a first quantity modulo a value at the modulus input; and
wherein the first quantity equals, modulo the value at the modulus input, a value at the base input raised to a power of a value at the exponent input.

4. (Previously Presented) An apparatus for generating an output signal indicating a multiplicative inverse of an integer modulo a prime modulus comprising:
a modulo exponentiation block configured to generate the output signal based on a first quantity modulo a value at a modulus input, the first quantity equal, modulo the value at the modulus input, to a value at a base input raised to a power of a value at an exponent input;
a first input for receiving a first signal indicating a value of the integer, the first input connected to the base input;
a second input for receiving a second signal indicating a value of the prime modulus, the second input connected to the modulus input; and

a circuit connected to the second input configured to generate on a first output a third signal indicating a value of the prime modulus less two, the first output connected to the exponent input.

5. (Previously Presented) An apparatus for performing a particular operation for using digital signatures on a network, the apparatus comprising a modulo exponentiation block configured for producing a multiplicative inverse of an integer modulo a prime modulus, wherein said multiplicative inverse is used in performing the particular operation.

6. (Previously Presented) The apparatus as recited in Claim 5, wherein the apparatus has no circuitry block configured to perform an extended Euclidian algorithm (EEA) and no general-purpose processor configured by instructions to perform the EEA.

7. (Original) The apparatus as recited in Claim 5, wherein:
the particular operation is performed in a series of sequential computations accomplished over a corresponding series of computation cycles; and
the apparatus further comprises connections configured to use the modulo exponentiation block during a plurality of computation cycles of the series of computation cycles.

8. (Previously Presented) The apparatus as recited in Claim 5, wherein the particular operation is a Rivest, Shamir, and Adleman encrypting operation.

9. (Previously Presented) The apparatus as recited in Claim 5, wherein the particular operation is a Rivest, Shamir, and Adleman decrypting operation.

10. (Original) The apparatus as recited in Claim 5, wherein the particular operation is a digital signature algorithm signing operation.

11. (Original) The apparatus as recited in Claim 5, wherein the particular operation is a digital signature algorithm verifying operation.

12. (Previously Presented) A computer-readable medium carrying one or more sequences of instructions for generating a multiplicative inverse of an integer modulo a prime modulus for use in performing a particular operation, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

sending data indicating a value of the integer as an base input to a modulo exponentiation function;

sending data indicating a value of the prime modulus as an modulus input to the modulo exponentiation function; and

sending data indicating a value of the prime modulus less two as an exponent input of the modulo exponentiation function,

wherein

the modulo exponentiation function generates an output based on a first quantity modulo the modulus input,

the first quantity equals, modulo the modulus input, the base input raised to a power of the exponent input; and

the output generated by the modulo exponentiation function is used in performing a particular operation that is selected from the group consisting of a digital signature algorithm signing operation, a digital signature algorithm verifying operation, an encryption operation for a first electronic message, and a decryption operation for a second electronic message.

13. (Original) The computer-readable medium recited in Claim 12, wherein the exponentiation function sends the base input, the modulus input and the exponent input to a special-purpose block of circuitry configured to perform modulo exponentiation.

14. (Previously Presented) A computer-readable medium carrying one or more sequences of instructions for generating a multiplicative inverse for use in determining a digital signature, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of

receiving and storing a first integer data value relating to a digital signature of an electronic message;

determining a multiplicative inverse of the first integer data value modulo a prime modulus data value by computing a first quantity modulo the prime modulus data value, wherein said computing includes using a modulo exponentiation block;

wherein the first quantity equals, modulo the prime modulus data value, the first integer data value raised to a power of a second quantity;

wherein the second quantity is two less than the prime modulus data value; and

storing the multiplicative inverse in a computer hardware storage element for use in determining the digital signature of the electronic message.

15. (Previously Presented) An apparatus for generating a multiplicative inverse for use in determining a digital signature, the method comprising the computer-implemented steps of:

means for receiving and storing a first integer data value relating to a digital signature of an electronic message;

means for determining a multiplicative inverse of the first integer data value modulo a prime modulus data value by computing a first quantity modulo the prime modulus data value, wherein said computing includes using a modulo exponentiation block;

wherein the first quantity equals, modulo the prime modulus data value, the first integer data value raised to a power of a second quantity;

wherein the second quantity is two less than the prime modulus data value; and

means for storing the multiplicative inverse in a computer hardware storage element for use in determining the digital signature of the electronic message.

16. (Previously Presented) An apparatus for generating an output signal indicating a multiplicative inverse of an integer data value modulo a prime modulus for use in performing a particular operation, the apparatus comprising:

means for sending a first signal, indicating a value of the integer data value, to a base input of a modulo exponentiation block of an electronic integrated circuit;

means for sending a second signal, indicating a value of the prime modulus, to a modulus input of the modulo exponentiation block; and

means for sending a third signal, indicating a value of the prime modulus less two, to an exponent input of the modulo exponentiation block;

wherein the modulo exponentiation block includes means for generating an output based on a first quantity modulo a value at the modulus input;

wherein the first quantity equals, modulo the value at the modulus input, a value at the base input raised to a power of a value at the exponent input; and

wherein the output generated by the modulo exponentiation block is stored in a computer hardware storage element for use in performing a particular operation that is selected from the group consisting of a digital signature algorithm signing operation, a digital signature algorithm verifying operation, an encryption operation for a first electronic message, and a decryption operation for a second electronic message.